Course Manual

`CAS CS 392 M1`: *Rust, In Practice and in Theory*

Boston University

Fall 2025

Contents

Course Code	`CAS CS 392 M1`
Course Title	Rust, In Practice and in Theory
Semester	Fall 2025
Instructors	Nathan Mull
Meeting Times	Tuesday and Thursday, 2:00PM-3:15PM
Meeting Location	MCS B37
Midterm Date	October 23 (during lecture)
Grade Breakdown	30% Assignments
	20% Midterm Exam
	40% Final Project
	10% Participation

Figure 1: Course overview

Abstract

This is a manual[1] for the course `CAS CS 392 M1`: *Rust, In Practice and in Theory*, which will be taught at Boston University during the Fall 2025 semester. It contains a general overview of the course and its policies. It does *not* contain specifics about the material being covered in the course; this appears on the course webpage. Figure 1 contains an overview of the course.

[1] Which takes the place of a syllabus.

Week 0 To-Do list

You should complete the following items within the first 48 hours of the start of the semester. Please reach out if you have concerns about any of the items listed.

- □ Verify that you have ready access to a laptop computer during the semester
- □ Verify that you know where the lecture is held
- □ Verify that you know where the discussion section in which you're registered is held
- □ Join Piazza with the following sign-up link
- □ Join Gradescope with the following sign-up link (entry code: 6K4D32)
- □ Familiarize yourself with the course webpage
- □ *(Optional)* Add the Piazza, Gradescope, and course webpage as bookmarks in your Internet browser
- □ Read this manual in its entirety

General Information

CAS CS 392 M1: *Rust, in Practice and in Theory* is a course about Rust, a type-safe, memory-safe programming language that is becoming a popular alternative to C and C++ in settings where performance and memory usage are major concerns. It's self-described as having "high-level ergonomics" and "low-level control." Practically speaking, this means clear, concise code with fewer memory bugs. Theoretically speaking, this means the use of a rich type system based on the notion of linearity to enforce memory-safety before any code has actually been run.

Despite its popularity, Rust is still daunting to learn, even for experienced programmers. There are several concepts in Rust that don't appear in any other popular languages. And even if you become a proficient Rust programmer, you may not have a deep understanding how Rust works, or why it is a better alternative to other low-level languages.

In this course, we'll spend the first half of the semester learning Rust. Our focus will be on the theoretical underpinnings of the language, which will mean we'll be looking at interpreters for subsets of Rust. We'll spend the second half building a project in Rust.

Disclaimer

Please read this section carefully. This is still a new course. It's the second time I'm offering it, it's on a quite modern topic and, to be frank, I'm not sure how well it's going to go. All of this is to say: if you're looking for a streamlined course experience, then I don't recommend taking this course. This course will require a level of academic independence and maturity that I expect of advanced undergraduate students. There's going to be experimentation, some things that work, some things that don't, some things with bad documentation, some things that will require you to look things up outside of the course material (gasp). But if you stick with the course, I hope it can be a valuable experience for all of us.

2.1 Prerequisites

The formal requirements of this course are:

- CAS CS 320: *Concepts of Programming Languages*
- CAS CS 210: *Computer Systems*

You'll have the best time with this course if you took CS320 during or after the Fall 2024 semester. It is not strictly a requirement, but if you didn't, then you'll have to catch up a bit on the type theory part of the course. We'll have a primer on this, but it will likely take some time and effort outside of class to master this material. Ideally, we should be able to create groups in which at least one member has this experience.

The dependence on `CS210` is quite mild. You need to be comfortable in the terminal, and you need to have some sense of low-level memory management. That said, the more versed you are in systems, the more you'll get out of learning Rust.

2.2 Learning Objectives

From this course we hope that you will:

- Learn the rudiments of Rust, a notoriously difficult but ultimately rewarding low-level programming language.
- Better understand the use-cases of Rust, when it works well, when it doesn't
- Identify the pitfalls of Rust programming and determine how to avoid them
- Learn to read/practice reading academic papers
- Learn to read/practice reading formal specifications of programming languages
- Implement interpreters for subsets Rust
- Prove properties like progress and preservation for a formal specification of Rust (properties which make Rust a "well-designed" programming language)

2.3 Course Structure

Lectures

I hold lectures each week on Tuesday and Thursdays (see the registrar and the course webpage for details). This course will be treated as a hybrid of a *flipped classroom* and a *workshop*. Each lecture will have corresponding readings which you will be expected to have read before lecture.

The first part of lecture will be dedicated to a short overview of the day's topic. I want you to think of this part of lecture as closer to your standard humanities course. It should be a high-participation discussion, with live-coding and board work. It's during this part that you should make sure you're on the same page as everyone else with respect to the material.

The second (primary) part will be dedicated to working with the material from the reading. During this part, you will be encouraged to work in groups of 2-4. Ideally, you'll be able to create a small group that you can work with consistently.

Lectures are mandatory. You're allowed to miss a handful of lectures due to conflicts, but the structure of this course only works if you to participate. If you feel like you can't do this (be it because of social anxiety, depression, family conflicts, etc.) then we can have a conversation and see what other options there are. But this does mean that you have to advocate for yourself.

2.4 Resources

The resources we'll use for this course are standard. Feel free to reach out directly if you have questions. All course material will be made available on the course webpage. Please check it frequently.

Programming

Most of the programming in this course is done in Rust. There will be some programming exercises done in OCaml(!) We'll set all of this up on the first day of the course. Please attend office hours and use Piazza if you need help troubleshooting. If you're are worried about access to technology, please contact me as soon as possible and we can see what we can do (though I cannot make any guarantees).

Course Communication

Course announcements and discussions will happen on Piazza. If you're unfamiliar with Piazza, see their support page for information and tutorials. Some policies regarding the use of Piazza. Make sure to set notifications correctly so you can keep up with updates regarding the course. "I didn't see the Piazza post about it" is never a valid excuse for missing a piece of information.

Submission

We'll be using Gradescope for assignment submissions. If you are unfamiliar with Gradescope, see their Get Started page for information and tutorials.

Evaluation

The grading breakdown for this course is given in Figure 3.1. The sites of evaluation are detailed in the following sections. Your raw percentage will be determined according to this breakdown and your final letter grade is guaranteed to be at least what is determined by Wheelock College's Grading Scale.[1] But, to borrow a phrase from Professor Mark Bun: "to correct for the possibility of [quizzes] and exams being more difficult than anticipated, letter grades may be (significantly) increased above these guarantees." Specifically, we may retroactively curve exam and quiz grades using a linear scale.[2]

Note. This is an upper-level undergraduate course with a small number of students. Generally speaking, the grading in this course tends to be less strict than in my larger courses. I'm primarily looking for participation and engagement with the material.

3.1 Assignments

Assignments are released (roughly) weekly on Thursdays during the first half of the semester and are due a week later on the following Thursday by 8:00PM. See the calendar on the course webpage for details. Assignments will consist primarily of programming tasks. total. I drop your lowest assignment grade, so only 5 assignments will count towards your final grade in the course. I will accept late assignments if you contact me 24hrs before a deadline to ask for an extension.

3.2 Midterm Exam

The midterm exam will be held on October 23 during lecture. It will be an in-class programming exam. It's meant to verify that you're prepared to program more heavily in Rust and to go further in depth into its concepts during the second half of the course.

[1] Formally we're part of the College of Arts and Sciences (CAS), but this grading scale is standard.

[2] See this article for details if you're interested.

30%	Assignments (7 total, 1 dropped)
20%	Midterm Exam
40%	Final Project (roughly 4 parts)
10%	Participation

Figure 3.1: Grade breakdown

3.3 Final Project

The bulk of your grade in this course will be determined by your final project. We'll say much more about this later in the semester, but in rough terms, you'll be working on month-long self-defined project in Rust or on a related topic. It will be ideally in groups of 2-3, but I am open to alternatives. We'll be in consistent communication about your project topic and progress in the second half of the semester.

You should think of the project as something which is a benefit to you and your portfolio. It should match your long-term interests, and should be something you'd like to include in an application or a CV. If you're interested in databases, then you could build a key-value store in Rust. If you're interested in type theory, you could do a survey on linear type theory. I'm open to almost any reasonable project, but I will also provide many possible example projects once we get to that part of the course, so don't be too worried if you're not sure what you want to do.

Policies

There are a number of policies associated with this course, some specific to the course and others which hold more generally in the university. These policies are detailed below in the following sections.

4.1 Diversity Statement

Our aim is to present material in a way that respects the diversity of the student body. If we fail to do this, please make me aware. Any suggestions are welcome and appreciated. I also expect students to appreciate and respect the unique opportunity they have to participate in a diverse student body like ours.

4.2 Disability Statement

If you require disability accommodations, please contact me as soon as possible. You should provide me with the appropriate documentation, available through the Disability and Access Services. In order to receive accommodations, you *must* be in contact with us.

If there's a policy that we're failing to comply with, please reach out with suggestions. And if you'd like accommodations that aren't covered by existing services or policies, feel free to contact me and we can see what we can do. I want everyone to feel able to fully participate in the course.

4.3 Sexual Misconduct

Please read the Sexual Misconduct Policy and review the entire page for information on talking to someone confidentially about experiences of sexual misconduct, filing a report, and any other relevant information. Above all, you should feel safe, and able to be productive. If this is not the case, please reach out to me or someone else immediately.

The members of the course staff are considered "mandated reporters" and are required to report cases of sexual misconduct. Therefore, **we cannot guarantee the confidentiality of a report**. I must provide our Title IX coordinator with relevant details such as the names of those involved in the incident. The university will consider a request for confidentiality and respect it to the extent possible.

With that in mind, if you come to any of me with questions or concerns, we will handle the situation to the best of our ability and connect you with available resources.

4.4 Academic Integrity

Please read the Academic Conduct Code and review the entire page for information about what constitutes academic dishonesty and what penalties arise as a result of violations of this code. This is taken very seriously at BU and I take it seriously in this courses. There are a couple policies about which I'll be strict:

- You must submit your own work for all assignments. Submitting the same file as another student, or something notably similar (e.g., identical wording or code in large parts of the solution) is considered academic misconduct and will be handled accordingly.
- Copying or information sharing regarding in-class evaluations like quizzes and exams is considered academic misconduct and will be handled accordingly.

If you work with others, consult materials found on the Internet, or use an AI assistant, you should cite your sources. This is a useful skill in any setting, and so we recommend being as conservative as possible regarding citations. In any assignment, these citations should appear next to every corresponding problem (in comments if the submission is code). Some examples:

- I discussed problem 1 and 2 with Leah Smith. She helped me understand X and Y aspects of the problem.
- I saw the stack overflow post stackoverflow.com/questions/6681284/python-numpy-arrays which informed my solution.
- I helped Zihan Guo with problem 4. I told them to try using X.
- I asked ChatGPT "what's the largest eigenvalue of this matrix?"

When in doubt, err on the side of longer, more descriptive citations. I do not consider missing or poor citations is a direct act of academic misconduct, but we will consider this grounds for further investigation in suspicious cases. Above all, use your best judgment and remember:

- I care about your success in this course. I provide a number of avenues to ask for help, please use them.
- You will have to answer questions on quizzes and exams without external aids (and in interviews when you apply for a job).
- If you don't know how to start thinking about a problem, it's okay to ask for pointers in office hours and on Piazza.
- I have safeguards (like dropped homework assignments) in the case you are unable to complete an assignment. In other words, don't submit someone else's work when you can drop an assignment.

4.5 Generative AI

The problem of generative AI in higher education will likely occupy us for the next decade or so. The role of these tools our lives is still an open question, one with many possible answers. But these tools exists, and the university, for better or for worse, has made them more accessible with the introduction of TerrierGPT, to which all students of the university have access. As such, all courses (including ours) are reviewing their policies. Keep in mind that this is all one big experiment. I don't know if our policy makes sense in the long term (or even now). But it's our attempt to come to terms with the appearance of these tools in our courses.

The policy this semester: *The use of generative AI are allowed to use these tools in assignments and projects* ***within reason****, but not in the midterm exam.* I understand that "within reason" is difficult to enforce, much of this will be on the honor systems. In an lower-division course, we might have a different policy, but a this point in your academic career, we expect that you are able to make the correct judgments as to the use of generative AI. Also, as you might have noted, we've re-weighted evaluation sites in order to put a greater emphasis on in-class closed-book evaluation.

An obligatory concluding remark: we know that existing models can solve many of the problems we will ask you. This does not negate the value in knowing how to do them without the help of external tools. To draw an imperfect analogy, we don't learn a new language in order to have memorized a vast collection of words and grammar rules, but in order to *internalize* the language, and learn how to *interact* with it and in it. This is our goal in this course and beyond. These tools can be incredibly useful in the process of learning and internalizing. But the internalizing is what we really want to achieve, so we can see a problem and can *sense* the underlying structure to which we can apply our knowledge from this course.

4.6 Additional Attendance Policies

As we've noted, we won't take attendance in our course. Instead, we remind you that, according to the Attendance policy at BU, you're required to attend the courses in which you're registered.

Absence Due to Religious Observance

According to the BU policy on Absence Due to Religious Observance: you "shall be excused from any such examination or study or work requirement, and shall be provided with an opportunity to make up such examination, study, or work requirement that may have been missed because of such absence on any particular day; provided, however, that such makeup examination or work shall not create an unreasonable burden upon such school."

Bereavement

According to the BU policy on Student Bereavement: you "should be granted up to five weekdays of bereavement leave for the death of an immediate family member." Your advisor should help you coordinate your leave.

4.7 Additional Grading Policies

Regrade Requests

Regrade requests may be submitted on Gradescope for up to one week after receiving the grade for an evaluation site. Regrade requests will only be considered in the case that the grader has made a mistake in grading. Any regrade requests which solely appeal for a higher grade will not be considered.

Grading Grievances

According to the BU policy on Grade Grievances: you may "contest a final course grade received in a unit-bearing Boston University course when that grade is alleged by the student to be arbitrary." Read the policy for more information. I recommend contacting me before submitting a formal appeal.

Incomplete Grades

According to the BU policy on Incomplete Coursework: "An incomplete grade (I) is used only when the student has conferred with the instructor prior to the submission of grades and offered acceptable reasons for the incomplete work. An incomplete grade may be appropriate when the student has participated in and completed requirements representing a majority of the course, and circumstances prevent the student from completing remaining requirements by the conclusion of the course.” In particular, **you must contact me before the last day of the semester in order to receive an incomplete grade.**

Closing Remarks

Quite a bit goes into organizing a course (even producing this document) as well as taking a course (even reading this document). In light of my comments on citations, I'll note that much of what's in this document is based on similar documents (often taken without permission) by Mark Crovella, Mark Bun, Jonathan Appavoo, Preethi Narayanan, Ravi Chugh, Andrew McNutt, and others I may be missing. All told, I hope that most of this logistical information will be overshadowed in your memory by the concepts themselves, and that we can focus on having a good time doing math and programming!

5.1 Course Agreement

In addition to a manual, I also consider this document a contract. The following is what you must agree to in order to remain in this course.

> *By enrolling in this course, I am agreeing to the policies outlined in this document, and I will uphold them to the best of my ability. I will also, generally speaking, try to be a reasonable person and be nice and good and respectful to the people around me taking—and running —the course. In return, I expect a high-quality learning experience and respect from those around me taking—and running—the course.*

5.2 University Resources

There are quite a few BU resources, it can sometimes be overwhelming. Here's a small list of the ones I think are important. If you're struggling in this course due to personal/health conditions, I can't guarantee I can help, but if you're comfortable reaching out, feel free to send me an email and I can see if I can point you towards the correct resources. If you're not comfortable reaching out to us, that's okay too, hopefully this list can help you find what you need. Also, keep in mind you can post anonymously on Piazza if you want to ask for help without including your name.

- Disability and Access Services
- Student Health Services
- Outreach and Prevention
- Behavioral Medicine
- Survivor Support (SARP)
- Educational Resources Center
- International Students & Scholars Office